

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-276387

(P2000-276387A)

(43) 公開日 平成12年10月6日 (2000. 10. 6)

(51) Int.Cl.	識別記号	F I	キーワード (参考)
G 0 6 F 12/00	5 2 0	G 0 6 F 12/00	5 2 0 E 5 B 0 7 5
	5 4 6		5 4 6 A 5 B 0 8 2
17/30		15/401	3 1 0 A
			3 3 0 Z

審査請求 未請求 請求項の数15 O L (全 16 頁)

(21) 出願番号 特願平11-77479

(22) 出願日 平成11年3月23日 (1999. 3. 23)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 金井 達徳

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72) 発明者 岐津 俊樹

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(74) 代理人 100083161

弁理士 外川 英明

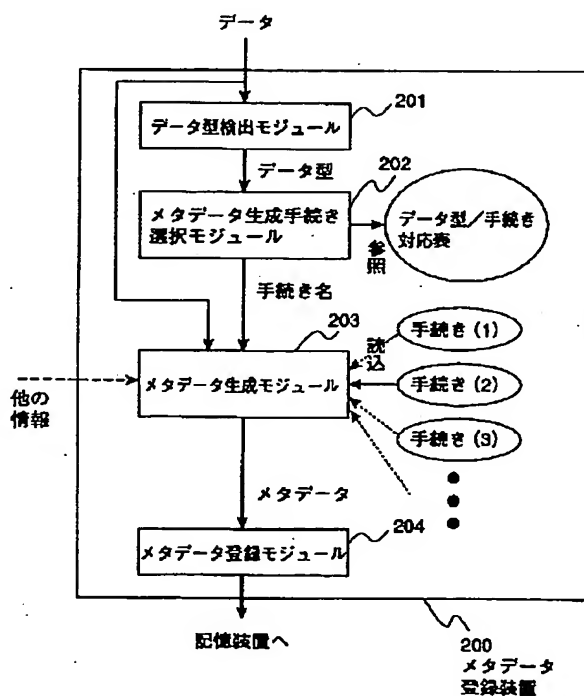
最終頁に続く

(54) 【発明の名称】 メタデータ登録方法および装置、データサーバ装置、並びにメタデータ登録用プログラムを記憶した記憶媒体

(57) 【要約】

【課題】 データサーバ装置においてメタデータを付与したデータを管理し高度な検索やアプリケーションプログラムの構築支援を実現するため、様々な種類の個々のデータに対して統一的なメタデータを付与する機構を提供する。

【解決手段】 データの型に応じたメタデータの生成手続き（プログラム）を登録しておき、データを登録する際にそのデータの型に対応したメタデータ生成手続きを使ってメタデータを自動的に付与する。



【特許請求の範囲】

【請求項 1】データを管理するデータサーバへ登録されるデータの型を検出し、

予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データの管理に使用可能なメタデータを生成するためのプログラムコードを有する手続きの中から、検出された型に対応する手続きを選択し、

選択された手続きのプログラムコードを実行してメタデータを生成し、

生成したメタデータを登録されるデータと関連付けて登録することを特徴とするメタデータ登録方法。

【請求項 2】前記複数の手続きは、それぞれが対応するデータの型に応じて生成すべきメタデータの種類を特定する情報を含み、この情報により特定されるメタデータの名前と値を生成するように作成されていることを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 3】前記複数の手続きは、アプリケーションもしくはユーザにとって同じ意味を持つメタデータの名前が、異なる手続きにより生成されるメタデータであっても、同じになるように作成されていることを特徴とする請求項 2 記載のメタデータ登録方法。

【請求項 4】前記メタデータの値を、前記登録されるデータの中に含まれる情報と、前記登録されるデータに含まれる情報に基づいて取得される別に記憶された情報のうち、少なくとも一方を抽出して生成することを特徴とする請求項 2 記載のメタデータ登録方法。

【請求項 5】前記検出は、前記登録されるデータのデータサーバへの登録要求に含まれる情報と、前記登録されるデータに含まれる情報と、前記登録されるデータの入力形態のうち、少なくともいずれかを検査することにより行うことを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 6】前記検出された型に対応する手続きが予め記憶された複数の手続きの中に存在しなかった場合に、手続きが存在しなかったことを示すメタデータを生成して前記登録されるデータと関連付けて登録する処理と、手続きが存在しなかったことをアプリケーションもしくはユーザに通知する処理のうち、少なくとも一方を行うことを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 7】前記複数の手続きのうち構造を有する文書に対応する手続きは、生成すべきメタデータの種類として構造を示す情報を含み、前記登録されるデータである文書中から該構造を示す情報により指定される部分を抽出し、抽出した部分を該手続きにより規定される名前と対応させてメタデータとして生成するように作成されていることを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 8】前記構造を有する文書に対応する手続きは、前記構造を示す情報を、文書構造を定義する定義情

報毎に規定し、アプリケーションもしくはユーザにとって同じ意味を持つメタデータの名前が、異なる定義情報に従う文書から生成されるメタデータであっても、同じになるように作成されていることを特徴とする請求項 7 記載のメタデータ登録方法。

【請求項 9】前記データサーバはディレクトリを用いてデータを管理するものであり、

前記データサーバへ新たなデータを登録するもしくは既存のデータを削除する場合に、予め記憶された複数のディレクトリ用手続きであって、それぞれが対応するディレクトリに応じて該ディレクトリの管理に使用可能なディレクトリ用メタデータを生成するためのプログラムコードを有する手続きの中から、登録もしくは削除されるデータが属するディレクトリに対応する手続きを選択し、

選択されたディレクトリ用手続きのプログラムコードを実行して新たなディレクトリ用メタデータを生成し、前記ディレクトリと関連付けて登録されたディレクトリ用メタデータを生成された前記新たなディレクトリ用メタデータに更新することを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 10】前記データサーバはディレクトリを用いてデータを管理するものであり、

前記登録されるデータについて生成されたメタデータが、前記ディレクトリの管理に使用されるディレクトリ用メタデータに適合しない場合に、前記登録されるデータのデータサーバへの登録要求を発したアプリケーションもしくはユーザに通知することを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 11】前記データサーバはディレクトリを用いてデータを管理するものであり、

前記登録されるデータについて生成されたメタデータが、前記ディレクトリの管理に使用されるディレクトリ用メタデータに適合する場合には、生成された該メタデータを登録せず、適合しない場合には、生成された該メタデータを前記登録されるデータと関連付けて登録することを特徴とする請求項 1 記載のメタデータ登録方法。

【請求項 12】前記データサーバは、前記登録されるデータを管理する際に、前記登録されるデータと関連付けて登録されたメタデータが存在すれば、該メタデータを用い、存在しなければ、前記ディレクトリ用メタデータを用いるものであることを特徴とする請求項 11 記載のメタデータ登録方法。

【請求項 13】データと該データについてのメタデータを登録し管理する管理手段を有するデータサーバへ登録されるデータを入力する手段と、

入力された前記データの型を検出する手段と、予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データについてのメタデータを生成するための手続きの中から、検出された前記型に

対応する手続きを選択する手段と、
選択された前記手続きにより生成された、入力された前記データについてのメタデータを、前記管理手段へ出力する手段とを備えたことを特徴とするメタデータ登録装置。

【請求項14】データの登録要求を受信する要求処理手段と、

この登録要求に基づいてデータを登録し、該データについてのメタデータとともに該データを管理する管理手段と、

前記データの型を検出する手段と、

予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データについてのメタデータを生成するためのプログラムコードを有する手続きの中から、検出された前記型に対応する手続きを選択する手段と、

選択された前記手続きのプログラムコードを実行させてメタデータを生成する手段と、

生成された前記メタデータを前記管理手段に登録させる手段とを備えたことを特徴とするデータサーバ装置。

【請求項15】データを管理するデータサーバへ登録されるデータに関連付けて登録すべきメタデータを生成する装置としての動作をコンピュータに実行させるためのメタデータ登録用プログラムを記憶した記憶媒体であって、

当該プログラムは、

データを管理するデータサーバへ登録されるデータの型を検出するための第一のプログラムコードと、

予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データの管理に使用可能なメタデータを生成するための手続きの中から、検出された型に対応する手続きを選択するための第二のプログラムコードと、

選択された手続きに従ってメタデータを生成するための第三のプログラムコードとを含むことを特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、データサーバ装置に用いられる、メタデータ登録装置および方法に関する。

【0002】

【従来の技術】コンピュータによる情報処理や、情報端末を介して利用者に情報を提供するためのシステムにおいて、データを登録・蓄積し、指定したデータを取り出したり、検索して所望の条件を満たすデータを取り出して使用するための装置として、データサーバ装置が使用される。データサーバ装置には、コンピュータのオペレーティングシステムの一部であるファイルシステムや、データベース管理システム、ネットワークを介してデー

タを出し入れできるWEBサーバやファイルサーバやデータベースサーバなどがある。

【0003】データサーバ装置のひとつであるWEBサーバは、図1に示すような木状の階層を持つ名前空間に様々なデータを管理する。この例では、`/proj/doc/fig/image1.gif`と`/proj/doc/fig/image2.gif`はGIF形式のイメージデータ、`/proj/doc/spec.doc`はワードプロセッサ(MS-WORD)で作った文書、`/proj/src/program.c`はC言語で書かれたプログラムのソースコード、`/sample/index.html`はHTMLで書かれた文書ファイルである。`/proj`や`/proj/doc/fig`で指定されるノードのように、データをまとめるためのノードは、一般にディレクトリと呼ばれる。ディレクトリには任意個のデータあるいはディレクトリを入れることが出来る。図1でトップレベルにある`/`は、ルートディレクトリと呼ばれる。

【0004】WEBサーバの管理するデータは、HTTPプロトコルによってネットワークを介してアクセスすることができる。HTTPプロトコルは、`"Hypertext Transfer Protocol -- HTTP/1.0"`(RFC1945)や`"Hypertext Transfer Protocol -- HTTP/1.1"`(RFC2068)に開示されている。例えばあるクライアントが`/proj/doc/fig/image1.gif`のデータを欲しい時には、`"GET /proj/doc/fig/image1.gif HTTP/1.1"`の形式のヘッダを持つGETリクエストをこのWEBサーバに送る。このリクエストを受けたWEBサーバは、要求されたデータを返す。

【0005】WEBサーバにデータを登録するために、同じくHTTPプロトコルのPUTリクエストを用いる。例えばあるクライアントが新しいGIF形式のイメージデータを`/proj/doc/fig/image3.gif`という名前で登録したい場合、`"PUT /proj/doc/fig/image3.gif HTTP/1.1"`という形式のヘッダを持つPUTリクエストをこのWEBサーバに送る。PUTリクエストの本体には、登録したいイメージデータを持たせる。このリクエストを受けたWEBサーバは、送られて来たイメージデータを指定された名前で格納する。

【0006】HTTPプロトコルは、GETやPUTでやり取りするデータの型を、MIMEタイプで指定する。例えばGIF形式のイメージデータの場合は、`"Content-type: image/gif"`というヘッダによって、転送するデータの型を指定する。

【0007】別のデータサーバ装置の例であるUNIXのファイルシステムも、図2に示すような木状の階層を

持つ名前空間に様々なデータ（ファイルシステムの場合はファイルと呼ぶ）を管理する。図2の例では、“`proc.c`”や“`sys.h`”等はファイル、“`src`”や“`include`”等はディレクトリと呼ばれる。ファイルシステム中のファイルをアクセスする時は、“`/src/kernel/proc.c`”のような階層を持ったファイル名でアクセスしたいファイル名を指定する。

【0008】データサーバ装置の管理するデータにメタデータを付けることで、データをより扱いやすくてできることが知られている。メタデータの例としては、データを作成した日付、場所、作成者、アクセス制御情報、更新記録、データソース（そのデータをどこから持ってきたかを示す情報）、チャンネル、EPG（電子番組表）、画像データの場合はサムネイル、GUIで表示するアイコン、画面上に表示する位置、そのデータを作成したアプリケーション、データをブラウズするためのアプリケーション、データのMIMEタイプ、データのサイズ、検索用のキーワードなど様々なものがある。

【0009】メタデータは、属性、プロパティ（property）、アトリビュート（attribute）などと呼ばれる場合もある。

【0010】データにメタデータをつけて管理することで、従来のように名前を指定してデータをアクセスするだけでなく、メタデータの値を指定して検索したり、アプリケーションで必要な情報をデータに関連付けて管理したり出来る。

【0011】WEBサーバにメタデータを持たせる方式は、WebDAVと呼ばれる仕様の一部として、“`HTTP Extensions for Distributed Authoring -- WEBDAV`”（INTERNET DRAFT: `draft-ietf-webdav-protocol-10`）に開示されている。WebDAVでは、WEBサーバの管理するデータにはプロパティと呼ぶメタデータを付ける事が出来る。

【0012】WebDAV対応のWEBサーバは、図3に示すようにデータを管理する。基本的には図1に示したWEBサーバと同じだが、通常のWEBサーバと違って、データとメタデータ（プロパティ）を関連付けて管理する。プロパティは名前と値のペアで保持される。図3の例では、“`/doc/fig/image1.gif`”で指定されるデータは、“`author`”、“`date`”、“`access`”の3つのプロパティを持っている。“`author`”プロパティはデータの作成者を値として持ち、この場合は“`T. Sato`”が作成者である。この例では、“`date`”はデータを作成した日付、“`access`”はアクセス権の情報を持っている。

【0013】WebDAVではディレクトリのことをコレクションと呼ぶ。図3で“`/`”や“`/doc/fig`”で指定される木の中間ノードはコレクションである。We

bDAVの仕様では、コレクションにもメタデータをつけることができる。以下の説明では、コレクションのことをディレクトリと呼ぶ場合もある。

【0014】プロパティをアクセスするために、WebDAVではHTTPプロトコルを拡張して新しいリクエストメソッドを追加している。プロパティの値を取り出すためにはPROPFINDメソッド、プロパティの値の設定あるいは削除のためにはPROPPATCHメソッドを用意している。これらのメソッドを使ったリクエストを出すことで、プロパティのアクセスが可能になる。PROPFINDやPROPPATCHのリクエストおよびリプライの本体は、XMLを使って記述するようになっている。

【0015】WebDAVの仕様に従ってプロパティを付けて管理されているデータを検索するための方式はDASLと呼ばれ、“`DAV Searching & Locating`”（INTERNET DRAFT: `draft-reddy-dasl-protocol-04.txt`）に開示されている。DASLは、HTTPプロトコルを拡張して検索用のSEARCHメソッドを追加し、SEARCHリクエストの本体にXMLで記述した検索条件を入れて送ると、その検索結果を同じくXMLで記述して送り返してくる。検索条件には、特定の名称のプロパティの値を指定したり、あるいはそれを複数のプロパティに対して組み合わせで指定したりできる。

【0016】UNIX型のファイルシステムにメタデータを持たせる例としては、“`Practical File System Design with the Be File System`”（Dominic Giampaolo, Morgan kaufmann Publishers, Inc. ISBN 1-55860-497-9）に開示されているBeファイルシステムがある。Beファイルシステムは、UNIXと同様の木状の階層構造の名称空間にファイルを管理し、各ファイルにアトリビュートと呼ぶメタデータを付けることができる。アトリビュートは名前と値のペアで、名前は文字列であり、値は文字列や整数、実数、バイナリなどのデータを持つことが出来る。

【0017】DASLと同様に、Beファイルシステムではアトリビュートの値あるいはその組み合わせによってファイルの検索をすることが出来る。

【0018】このように、メタデータを管理できるデータサーバ装置においては、メタデータとして様々な情報をデータに関連付けて管理することができる。また、データあるいはファイルの名前による単純なアクセスではなく、様々な情報をキーとする検索によるより高度なアクセスが可能になる。さらに、アプリケーションがデータに関連する情報をメタデータとして関連付けて管理することで、アプリケーションプログラムの構築も容易に

なる。

【0019】

【発明が解決しようとする課題】このように、データサーバ装置において、メタデータをデータに関連付けて管理することは、さまざまな利点を持っているが、実際にそれぞれのデータにメタデータを付けなくては其の利点を生かすことができない。しかし、メタデータを個々のデータに対して付けるためには非常に煩雑な作業が必要になる。

【0020】ファイルの作成日やサイズなどのように、データの登録時に自動的に付けられるメタデータもある。しかしこれは、データの型などに依存しない汎用的なメタデータに限られる。それ以外のメタデータに関しても、何らかの手段でメタデータを付ける作業をある程度でも自動化できれば、ユーザは煩雑な作業から解放されることが期待できる。

【0021】さらに、メタデータを付ける際に、付ける人や付ける時によって、同じ意味を持つメタデータにバラバラな名前をを付けたり、同じ型のデータに対してそれを扱うアプリケーションがそれぞれ勝手な名前でメタデータを付けると、後で有効な検索が出来なくなる。ある程度統一的にメタデータを付ける手段が望まれる。

【0022】本発明は、このような事情に鑑み、データサーバ装置においてメタデータを利用した高度な検索やアプリケーションプログラムの構築支援を実現するために、様々な種類の個々のデータに対して統一的なメタデータを付与する機構を提供することを目的とする。

【0023】

【課題を解決するための手段】本発明は、それぞれデータの型に応じたメタデータの生成手続き（プログラム）を複数、コンピュータ（メタデータ登録装置もしくはデータサーバ装置）に記憶しておき、この中から、データサーバ装置に登録するデータの型に対応したメタデータ生成手続きを選択し、これを用いてメタデータを自動的に生成し、データに付随させて登録する。

【0024】本発明に係るメタデータ登録方法は、データを管理するデータサーバへ登録されるデータの型を検出し、予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データの管理に使用可能なメタデータを生成するためのプログラムコードを有する手続きの中から、検出された型に対応する手続きを選択し、選択された手続きのプログラムコードを実行してメタデータを生成し、生成したメタデータを登録されるデータと関連付けて登録することを特徴とする。

【0025】ここで、前記複数の手続きは、例えば、それぞれが対応するデータの型に応じて生成すべきメタデータの種類を特定する情報を含み、この情報により特定されるメタデータの名前と値を生成するように作成されている。

【0026】さらに、アプリケーションもしくはユーザ

にとって同じ意味を持つメタデータの名前が、異なる手続きにより生成されるメタデータであっても、同じになるように作成されていると、統一的なメタデータが付与できる。

【0027】前記メタデータの値は、前記登録されるデータの中に含まれる情報と、前記登録されるデータに含まれる情報に基づいて取得される別に記憶された情報のうち、少なくとも一方を抽出して生成すると良い。

【0028】また、前記検出は、前記登録されるデータのデータサーバへの登録要求に含まれる情報と、前記登録されるデータに含まれる情報と、前記登録されるデータの入力形態のうち、少なくともいずれかを検査することにより行うことができる。

【0029】前記検出された型に対応する手続きが予め記憶された複数の手続きの中に存在しなかった場合には、手続きが存在しなかったことを示すメタデータを生成して前記登録されるデータと関連付けて登録する処理と、手続きが存在しなかったことをアプリケーションもしくはユーザに通知する処理のうち、少なくとも一方を行うこととしても良い。

【0030】前記複数の手続きのうち構造を有する文書（例えばXML、HTML、SGML等）に対応する手続きは、生成すべきメタデータの種類として構造を示す情報（例えばエレメント名等）を含み、前記登録されるデータである文書中から該構造を示す情報により指定される部分を抽出し、抽出した部分を該手続きにより規定される名前と対応させてメタデータとして生成するように作成されていることが好ましい。

【0031】さらに、前記構造を有する文書に対応する手続きは、前記構造を示す情報を、文書構造を定義する定義情報（例えばDTD等）毎に規定し、アプリケーションもしくはユーザにとって同じ意味を持つメタデータの名前が、異なる定義情報に従う文書から生成されるメタデータであっても、同じになるように作成されていると、統一的なメタデータが付与できる。

【0032】前記データサーバがディレクトリを用いてデータを管理するものであるシステムに本発明を適用する場合の、特徴的な例を幾つか挙げる。

【0033】一つ目の例では、前記データサーバへ新たなデータを登録するもしくは既存のデータを削除する場合に、予め記憶された複数のディレクトリ用手続きであって、それぞれが対応するディレクトリに応じて該ディレクトリの管理に使用可能なディレクトリ用メタデータを生成するためのプログラムコードを有する手続きの中から、登録もしくは削除されるデータが属するディレクトリに対応する手続きを選択し、選択されたディレクトリ用手続きのプログラムコードを実行して新たなディレクトリ用メタデータを生成し、前記ディレクトリと関連付けて登録されたディレクトリ用メタデータを生成された前記新たなディレクトリ用メタデータに更新する。

【0034】二つ目の例では、前記登録されるデータについて生成されたメタデータが、前記ディレクトリの管理に使用されるディレクトリ用メタデータに適合しない場合に、前記登録されるデータのデータサーバへの登録要求を発したアプリケーションもしくはユーザに通知する。

【0035】三つ目の例では、前記登録されるデータについて生成されたメタデータが、前記ディレクトリの管理に使用されるディレクトリ用メタデータに適合する場合には、生成された該メタデータを登録せず、適合しない場合には、生成された該メタデータを前記登録されるデータと関連付けて登録する。このとき、前記データサーバは、前記登録されるデータを管理する際に、前記登録されるデータと関連付けて登録されたメタデータが存在すれば、該メタデータを用い、存在しなければ、前記ディレクトリ用メタデータを用いるものであることが好ましい。

【0036】本発明に係る各方法の発明は、それぞれ装置の発明としても成り立つ。

【0037】例えば、本発明に係るメタデータ登録装置は、データと該データについてのメタデータを登録し管理する管理手段を有するデータサーバへ登録されるデータを入力する手段と、入力された前記データの型を検出する手段と、予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データについてのメタデータを生成するための手続きの中から、検出された前記型に対応する手続きを選択する手段と、選択された前記手続きにより生成された、入力された前記データについてのメタデータを、前記管理手段へ出力する手段とを備えたことを特徴とする。

【0038】また、本発明に係るデータサーバ装置は、データの登録要求を受信する要求処理手段と、この登録要求に基づいてデータを登録し、該データについてのメタデータとともに該データを管理する管理手段と、前記データの型を検出する手段と、予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データについてのメタデータを生成するためのプログラムコードを有する手続きの中から、検出された前記型に対応する手続きを選択する手段と、選択された前記手続きのプログラムコードを実行させてメタデータを生成する手段と、生成された前記メタデータを前記管理手段に登録させる手段とを備えたことを特徴とする。

【0039】本発明に係る各方法の発明はまた、それぞれプログラムを記録した記録媒体の発明としても成り立つ。

【0040】例えば、本発明に係る記憶媒体は、データを管理するデータサーバへ登録されるデータに関連付けて登録すべきメタデータを生成する装置としての動作をコンピュータに実行させるためのメタデータ登録用プログラムを記憶した記憶媒体であって、当該プログラム

は、データを管理するデータサーバへ登録されるデータの型を検出するための第一のプログラムコードと、予め記憶された複数の手続きであって、それぞれが対応するデータの型に応じて該データの管理に使用可能なメタデータを生成するための手続きの中から、検出された型に対応する手続きを選択するための第二のプログラムコードと、選択された手続きに従ってメタデータを生成するための第三のプログラムコードとを含むことを特徴とする。

【0041】

【発明の実施の形態】以下、図面を参照しながら本発明の実施形態について説明する。

【0042】一般にデータサーバ装置100は図4のような構成を持つ。記憶装置103は、磁気ディスク等の実際にデータを記録する装置である。データ管理部102は、どのデータが記憶装置のどの領域に記録されているかを管理する。アクセス要求処理部101は、アプリケーションプログラムあるいはユーザ（ネットワーク上のターミナルデバイス、クライアント等を含む）であるアクセス要求元からのアクセス要求を受け、そのアクセスに必要なデータの読み書きをデータ管理部102に指示し、要求処理結果をアクセス要求元に返す。ここで要求処理結果とは、データの書き込み要求（例えばデータを含むPUT）の場合には書き込みが成功したか失敗したかのステータス情報（レスポンス）であり、データの読み出し要求（例えばリクエストを示すGET）の場合には読み出したデータ（あるいは読み出し失敗を知らせるステータス情報）である。

【0043】本実施形態に係るメタデータ登録装置200は、例えば図5に示す構成を持つ。このメタデータ登録装置200を備えたデータサーバ装置300は、例えば図6に示す構成を持つ。なお、メタデータ登録装置及びデータサーバ装置の詳細はそれぞれ図5及び図4に示し、図6ではこれらの個々の構成は簡略化して示している。

【0044】データサーバ装置300にデータを登録する要求（例えばPUT）が来ると、データサーバ装置のアクセス要求処理部301は、そのデータをデータ管理部302に渡して記憶装置303への登録処理を行うと共に、そのデータをメタデータ登録装置202のデータ型検出モジュール201にも渡す。

【0045】データ型検出モジュール201は、アクセス要求処理部301から受け取ったデータの型を調べる。

【0046】メタデータ生成手続き選択モジュール202は、様々なデータ型とその型のデータのメタデータを生成する手続き（プログラム）との対応を、例えば表の形で管理している。そして、データ型検出モジュール201が検出したデータの型を受け取り、上記対応表を参照して、そのデータ型に対応するメタデータ生成手続き

を選択する。

【0047】メタデータ生成モジュール203は、メタデータ生成手続き選択モジュール202が選択したメタデータ生成手続きに対応するプログラムを読み込み、これを用いて、アクセス要求処理部301から渡されて来たデータに対するメタデータを生成する。

【0048】メタデータ登録モジュール204は、メタデータ生成モジュール203において生成したメタデータを、データ管理部302に指示することにより、該当するデータと関連付けて登録する。

【0049】なお、上記の各部や各モジュールは、ソフトウェアとしても、ソフトウェアとハードウェアの組み合わせとしても実現できるものである。また、汎用のコンピュータに、上記の各モジュールの機能を実行させるためのプログラムをインストールすることで、本発明を実施することができる。

【0050】ここでは、WebDAV対応のWEBサーバに本発明を適用する例を説明する。この実施例では、WEBサーバに登録するデータのMIMEタイプやデータの名前の拡張子から、データの型を検出する。

【0051】本実施例に係るWEBサーバは、図7に示すようなメタデータ生成手続き表（データ型／手続き対応表）を持つ。メタデータ生成手続き表は、MIMEタイプ名と、そのMIMEタイプに対応するデータの拡張子名と、そのMIMEタイプに対応するメタデータ生成手続き名の3種類の情報の対応表である。例えば、MIMEタイプが“image/jpeg”のデータの拡張子は“jpeg”あるいは“jpg”で、そのメタデータ生成手続きは“meta jpeg”で指定されるものである。ここでメタデータ生成手続き名は、実際には関数へのポインタあるいは実行すべきプログラムモジュール名である。

【0052】このようなメタデータ生成手続き表を用意するのではなく、メタデータ登録プログラム（汎用コンピュータを、仮想的に図5に示すような構成を持つメタデータ登録装置として、動作させるための、プログラム）中に条件分岐としてこの対応関係を埋め込む実施法も可能である。

【0053】本実施例のWEBサーバがデータの登録要求であるPUTリクエストを受け取ると、そのデータを指定された名前前で登録すると共に、図8に示すようなメタデータ登録処理を開始する。まず、PUTリクエストのヘッダに登録したいデータのMIMEタイプが指定されているかどうか調べる（ステップS1）。

【0054】MIMEタイプが指定されていれば、そのMIMEタイプ名をキーにしてメタデータ生成手続き表を引き、そのMIMEタイプのデータののためのメタデータ生成手続き名を取り出す（ステップS2）。MIMEタイプが指定されていない場合には、登録するデータの名前から拡張子の部分を取り出し、その拡張子をキーに

してメタデータ生成手続き表を引き、その拡張子を持つデータののためのメタデータ生成手続き名を取り出す（ステップS3）。このとき、ステップS2で登録するデータのMIMEタイプがメタデータ生成手続き表に存在しない場合、あるいはステップS3で登録するデータの拡張子がメタデータ生成手続き表に存在しない場合には、このデータののためのメタデータ生成手続きは用意されていないことを示している（ステップS4No）、メタデータを付けずに終了する。メタデータ生成手続きが見つければ（ステップS4Yes）、その手続きを実行して、登録するデータに付けるべきメタデータを生成する（ステップS5）。

【0055】ここで生成したメタデータは、例えば図9に示すように、メタデータ名と値のペアが任意の個数集まったものである。この例では、“date”という名のメタデータの値は“1998-07-12”、“author”という名のメタデータの値は“Y. Suzuki”、“program”という名のメタデータの値は“photoshop4.0”である。こうしてメタデータを生成した後、登録するデータに関連付けてメタデータを登録する（ステップS6）。

【0056】この実施例では、登録するデータの型に対応するメタデータ生成手続きが見つからなかった場合、ステップS4において何もせずに終了しているが、それ以外にも次のような実施法がある。すなわち、そのデータに対してメタデータ生成手続きが見つからなかったことを示すメタデータをステップS5にて生成し、これをステップS6にて登録対象のデータに関連付けて登録するようにしても良い。あるいは、登録対象のデータに対応したメタデータ生成手続きが見つからないのでメタデータの自動生成はしないことを、データを登録するユーザに通知するようにしても良い。この通知によって、ユーザは、その登録データにはメタデータが付与されておらず、その後のメタデータを用いた検索ではその登録データを探し出せないということを、認識することができる。また、誤ったデータを登録したり、データの登録方法を間違えたりした場合に、ユーザにその旨を知らせる効果もある。

【0057】ここではデータ型の検出法として、PUTリクエストのヘッダに指定されたMIMEタイプを使用する方式と、登録するデータの名前の拡張子を元に決定する方式の2種類の方式を、MIMEタイプの方を優先させて併用する例を説明した。他にも、拡張子の方を優先させて併用するのも良いし、MIMEタイプのみ使用するのでも、拡張子のみ使用するのでも、構わない。また、一旦MIMEタイプに基づいてステップS2で選択したメタデータ生成手続きをステップS5で実行した結果、メタデータの生成が適切に行えなかったことが判明した場合には、今度は拡張子に基づいてメタデータ生成手続きを選択し直し、改めてこれを実行してメタデー

タを生成するようにすることも可能である。

【0058】また、データ型の検出法として、他に、データの先頭の数バイトを調べて、その値によって型を判断する方式も採用可能である。あるいは、データの先頭ではなく、データの中を調べることによって、データの型を検出することもできる。例えばMPEGのデータのような場合、そのデータ中に特定のパターンのデータシーケンスがあるかどうかを調べることによって、そのデータがMPEGデータという型であることが判断できる。

【0059】またここでは、登録するデータの型に対応したメタデータ生成手続きが、図9に示すような決まった形式で付けるべきメタデータの一覧を作り、それを受け取った共通の処理ステップ（図8のステップS6）で登録するデータにメタデータを付けるようにする例を説明したが、この他にも、メタデータ生成手続きが登録するデータに直接メタデータを付けるように実装することも可能である。

【0060】メタデータ生成手続き（プログラム）は、それぞれのデータの型に応じて予め作成してコンピュータ（データサーバ装置／メタデータ登録装置）に備えておくようにする。例えば、ワープロなどのアプリケーションの文書データは、その文書データ中に、そのアプリケーションに応じたフォーマットで、文書の作成日やタイトル、作成者などの情報を保持している。予め調べた文書データのフォーマットに合わせて、そのような情報を取り出してメタデータとするように、メタデータ生成手続きを作成する。電子文書の流通に広く利用されているPDFのようなフォーマットの場合も同様である。

【0061】広く用いられているEXIF規格のデジタルカメラのデータは、“デジタルスチルカメラ用画像ファイルフォーマット規格（Exif）Version 2.1”（JEIDA-49-1998、日本電子工業振興協会）に開示されているように、撮影に用いたカメラの機器名、撮影日時、撮影者、画像のタイトル、シャッタースピード、絞り値、レンズ焦点距離、撮影場所の緯度・経度・高度、コメントなど、様々な情報を内部に持っている。EXIF規格のデータは、MIMEタイプとしては“image/jpeg”あるいは“image/tiff”であり、これらの型に対応するメタデータ生成手続きの中で、JPEGあるいはTIFFフォーマットのファイル中にEXIFの情報の領域が含まれているかどうか調べ、含まれていればEXIF情報をメタデータとして取り出すように実装する。含まれていなかった場合でも、解像度などのJPEGあるいはTIFFフォーマット固有のメタデータは取り出すことが出来る。

【0062】広く用いられているDV規格のデジタルビデオのデータは、“Recording—Helical—scan digital video cas-

ette recording system using 6.35mm magnetic tape for consumer use (525-60, 625-50, 1125-60 and 1250-50 systems) —”（IEC 61834）に開示されているように、ビデオストリーム中に、録画した時間、録画内容を示すカテゴリ、言語、放送番組を録画した場合のチャンネル、録画した場所の緯度・経度、カメラの各種設定や機材データなど、様々な情報を内部に持っている。DV規格のデータは、パソコン上で広く使われているAVI形式やQuickTime形式のファイルとして記録されることが多く、MIMEタイプとしては“video/x-msvideo”あるいは“video/quicktime”であり、これらの型に対応するメタデータ生成手続きの中で、AVIあるいはQuickTimeフォーマットのファイル中にDVのデータが含まれているかどうか調べ、含まれていればDVの持つ様々な情報をメタデータとして取り出すように実装する。

【0063】メタデータ生成手続きは、単に登録すべきデータ中に含まれている情報をメタデータとして取り出すだけでなく、登録すべきデータ中には含まれない情報を外部から獲得してメタデータとして付けることも可能である（図5のメタデータ生成モジュール203に入る他の情報の点線矢印の一例に相当）。例えば、DV規格のビデオデータの場合、そのデータ中に含まれている録画時間と放送チャンネルの情報を元に、その番組に関する情報を電子番組表（EPG）から取り出し、それに含まれる番組タイトルや出演者などの情報をメタデータとして登録するように実施することが出来る。電子番組表の情報を取り出す手段は、放送電波の一部を利用して送られて来る電子番組表情報を、本データサーバ装置内に、もしくは本データサーバ装置と接続されたコンピュータ内に記憶しておき、その中から該当する番組の情報を取り出すように実施出来る。あるいは、インターネットなどの通信回線を介して、予め登録しておいた別のサーバから、該当する番組の電子番組表情報を取り出すようにも実施できる。

【0064】メタデータ生成手続きによって生成するメタデータの種類や数は、データの型によって様々に異なっていて構わない。しかし、例えば日付や場所、タイトルやデータの作成者など、データの型は違って同じ意味を持つメタデータが多くある。このようなメタデータについては、異なる型のデータから取り出した情報を同じ名前のメタデータとして登録するように、各メタデータ生成手続きを作り込んでおくことが好ましい。すなわち、例えば、文書データから取り出した作成日付も、JPEGデータのEXIF情報から取り出した撮影日付も、メタデータ名を“date”として登録するように、文書データに対応するメタデータ生成手続きと、JPE

Gデータに対応するメタデータ生成手続きとを作成しておく。

【0065】このように実装することで、異なる型のデータが混在する場合にも意味のある検索が可能になる。例えば、同じ日に作成した、EXIF形式のデジタルカメラのデータやDV形式のデジタルビデオのデータ、更にはスケジュールデータなどが混在している時に、例えば日付を指定して検索すると、その日のスケジュールとその日に撮影した写真とビデオを選び出してくることが出来る。他にも、カメラでもビデオでも何でも良いからある場所に関するデータを探したいという検索なども可能になる。

【0066】登録するデータの型がXML文書の場合には、さらにそのXML文書の内容に応じてメタデータの生成の仕方を変えるように実施することが出来る。

【0067】XML文書は図10に示すようにエレメントと呼ばれる階層的なタグ付き木構造を持つ。すなわち、“<book>”と“</book>”で囲まれるのがエレメント名“book”のエレメントであり、このエレメントはその中に“<title>”と“</title>”で囲まれるエレメントを始めとして複数のエレメントを持つ。XML文書にはDTDと呼ばれる文書の定義情報を関連付けることができる。図10の例では、“book.dtd”というDTDを関連付けている。DTDには、その文書の中にはどのようなエレメントがどのような順序あるいは階層で出現するかが定義される。“book.dtd”の場合には、“book”というトップレベルのエレメントの中には、“title”、“author”、“publisher”などのエレメントが並ぶといった定義がされている。XMLの文書は必ずDTDを持つわけではなく、DTDが無い場合もある。この場合、どのようなエレメントがどのような順序あるいは階層で使われるかはまったく自由である。XML文書の構造を定義する手段としては、DTD以外にもDDML、DCD、SOX、XML-DATA、RDFなどさまざまな方式があるが、本発明を適用するメタデータ登録装置では、いずれの方式でも、DTDの場合と同様に扱える。

【0068】本実施例のWEBサーバは、PUTリクエストによるデータの登録要求を受け取ると、メタデータ生成手続き選択モジュール202が、データ型／手続き対応表を引くことにより、MIMEタイプが“text/xml”と指定されているか、あるいは拡張子が“xml”であれば、XML文書であると判断する。次に、メタデータ生成モジュール203が、XML文書というデータ型に対応するメタデータ生成手続きを選択的に実行するが、このXML文書用のメタデータ生成手続きは、(図示しない)XMLメタデータ対応表を引くことにより、登録するXML文書の中にXMLメタデータ対応表に指定されたエレメントが存在するかどうかを調べ、存

在すればそのエレメントの持つ情報をメタデータとして抽出する。

【0069】XMLメタデータ対応表は、例えば図11に示すように、DTD名とエレメント名と対応するメタデータ名の3つのデータ項目からなる表として実施できる。図10に示したXML文書の場合、DTD名は“book.dtd”であり、それを使って図11のXMLメタデータ対応表を引くと、その中の“title”という名前のエレメントの持つ情報を“title”という名前のメタデータの値に、“author”という名前のエレメントの持つ情報を“author”という名前のメタデータの値に、“keywords”という名前のエレメントの持つ情報を“keyword”という名前のメタデータの値にする。その結果、図12に示すようなメタデータを生成することが出来る。

【0070】XML文書用のメタデータ生成手続きによって生成するメタデータの種類や数は、登録対象のXML文書によって様々に異なっていて構わない。しかし、例えば日付や場所、タイトルやデータの作成者など、各XML文書中のエレメント名は違っても同じ意味を持つメタデータについては、異なるエレメント名のエレメントから取り出した情報を同じ名前のメタデータとして登録するように、XMLメタデータ対応表を作っておくことが好ましい。すなわち、例えば、図11では、“photo.dtd”から取り出した“creator”の情報も、“book.dtd”から取り出した“author”の情報も、メタデータ名を“author”として登録するようになっている。このように実装することで、異なるDTDのXML文書が混在する場合にも意味のある検索が可能になる。

【0071】この実施例では、XMLメタデータ対応表の中で、メタデータとして抽出すべき情報をDTD名とエレメント名とによって指定している。メタデータとして抽出すべき情報を、エレメント名だけで指定するように実施することも出来る。この場合、指定された名前のエレメントを持っていればどのようなXML文書からでも、それに対応するメタデータを抽出する。DTDを持たないXML文書の場合も同様である。

【0072】あるいは、エレメント名だけではなく、ネームスペース名とエレメント名を組にして、メタデータとして情報を抽出すべきエレメントを指定するように実施することもできる。ネームスペース名とは、同じエレメント名であってもどのDTDで定義されたかによって異なる意味を持つ場合を区別するために、付与されるものである。

【0073】さらに、メタデータとして抽出するようにXMLメタデータ対応表に指定されたエレメントの持つ情報を、そのままメタデータの値として登録するのではなく、その情報から不要なエレメントのタグ(指定されたエレメントが更に別のエレメントを含む場合のその別

のエレメントのタグ等)を抜いたり、情報の内容を要約したり、日付などは何らかの標準的な形式に変換したりした後、メタデータとして登録するように実施することもできる。

【0074】XML文書のMIMEタイプとしては、“application/xml”が使われる場合もあるが、この場合も上述した例と同様に実施できる。

【0075】本実施形態に係るメタデータ登録装置を用いてメタデータを生成・登録するタイミングは、WEBサーバ、データベースサーバ、ファイルサーバ等のデータサーバ装置がデータ登録要求を受け取って処理する時点になるように実施することも出来るし、データ登録後に非同期に、新しく登録されたデータのメタデータを生成・登録するように実施することも出来る。

【0076】本実施形態に係るメタデータ登録装置は、メタデータを管理できるファイルシステムにも同様に適用できる。WEBサーバではデータの登録はPUTリクエストでデータ全体を一括して登録するようになるが、ファイルシステムの場合は、既存のデータの一部分が更新されるような場合にも対処できることが望ましい。そのため、ファイルのクローズ時にWEBサーバの場合と同様のメタデータ生成・登録処理を行うようにすると良い。また、ファイルのクローズ時にはどのファイルが更新されたかを記録しておき、後で非同期に、更新されたファイルを探してそれらに対しメタデータ生成・登録処理を行うように実施することも出来る。

【0077】本実施形態に係るメタデータ登録装置では、登録されるデータの型に基づいて、その型に適したメタデータ生成手続きを選択する。データの型の判断の方式として、これまでの実施例で述べた方式以外にも、特定の入力装置から入って来たデータは予め定めておいた型として扱う方式を採用することも出来る。例えば、IEEE1394ネットワーク経由でデジタルビデオから送られて来るビデオデータは常にDV形式であると仮定してDV形式の型のデータからメタデータを抽出する手続きを実行するように実施することも出来る。

【0078】データだけでなくディレクトリにもメタデータを付与できるシステムの場合、データ登録時に、そのデータに対するメタデータを生成するだけでなく、そのデータを登録するディレクトリのメタデータを更新する機能を設けると良い。すなわち、ディレクトリがその中に持つデータや子ディレクトリは、新しいデータの登録や削除によって変化する。そのため、通常のデータのように登録時にメタデータを付けるだけではなく、中に持つデータや子ディレクトリが追加/削除される度にメタデータの変更の必要性を調べ、必要であればディレクトリのメタデータを更新する。

【0079】ディレクトリ内のデータや子ディレクトリの追加/削除時に、そのディレクトリのメタデータとして更新する情報には、各データおよびメタデータの持つ

日付の範囲、同様に場所の範囲、同様に検索用のキーワードなどがある。すなわち、ディレクトリのメタデータには、そのディレクトリ内のデータや子ディレクトリの各メタデータに共通の特徴を表す情報を含むようにする。

【0080】ディレクトリのメタデータを付けるために必要になるメタデータ生成手続きは、データの型毎に用意したメタデータ生成手続きと違って、ディレクトリと対応付けて管理する必要がある。この対応付けの手段としては、ディレクトリ名とそのディレクトリのためのメタデータ生成手続きの組を対応表として管理する方式がある。また別の手段としては、そのディレクトリの中にある特定の名前のデータとして、そのディレクトリのメタデータ生成手続きの情報が入っているように実施する方式がある。さらに別の手段としては、そのディレクトリの持つ特定の名前のメタデータの値が、そのディレクトリのメタデータ生成手続きあるいはそれを得るための情報になっているように実施する方式がある。これらは単独で用いても良いし、組み合わせで用いることも出来る。複数の手段を、予め定めておいた順に適用可能かどうか調べ、最初に適用可能である手段を用いるように実施することも出来る。

【0081】具体的な動作を、上記のうちの最初の方式を対応付けの手段として採用する場合を例にとり、説明する。アクセス要求処理部301もしくはデータ管理部302は、データや子ディレクトリの追加/削除が起こると、これをメタデータ生成手続き選択モジュール202へ通知する。メタデータ生成手続き選択モジュール202は、追加/削除されるデータもしくは子ディレクトリを収容するディレクトリに対応付けられたメタデータ生成手続き(プログラム)を、ディレクトリ名/手続き対応表を参照して選択する。メタデータ生成モジュール203は、選択された手続きを読み込んで実行する。例えば、そのディレクトリのメタデータを記憶装置303からデータ管理部302経由で読み出してその内容を調べるとともに、そのディレクトリの中のデータもしくは子ディレクトリ(追加/削除されるものと、その他のもののうち、少なくとも一方)のメタデータを読み出してその内容を調べ、追加/削除によりそのディレクトリ内のデータや子ディレクトリの各メタデータの共通の特徴が変化する場合には、ディレクトリの新たなメタデータを生成する。そして、メタデータ登録モジュール204が、ディレクトリのメタデータを新たに生成した内容に書き換える。

【0082】その他にも、メタデータ生成手続きが登録したいデータのメタデータを生成する際に、それを登録するディレクトリのメタデータを参照して(図5のメタデータ生成モジュール203に入る他の情報の点線矢印の他の例に相当)、メタデータの生成の仕方を変えるように実施することも出来る。

【0083】例えば、登録するデータから生成したメタデータと同じ名前のメタデータをディレクトリが見つかどうかを調べる。持っていた場合に、登録したいデータのメタデータの値とディレクトリのメタデータの値を比べて、違っている（一致しない、もしくは、ディレクトリの方の値の範囲に登録したいデータの方の値が入らない）場合には、データの登録要求元にその事を知らせる。このように実施する事により、例えば同じ日付のデータだけを入れたいディレクトリ内に、異なる日付のデータを誤って登録することを防止する効果がある。

【0084】また別の例としては、登録するデータから生成したメタデータと同じ名前のメタデータをディレクトリが見つかどうかを調べる。持っていた場合に、登録したいデータのメタデータの値とディレクトリのメタデータの値を比べて、同じであれば登録するデータにはそのメタデータを生成しない。異なっている場合には、登録するデータにそのメタデータを生成する。これにより、あるデータに値を知りたいメタデータが付いていない場合には、そのデータが存在するディレクトリの持つ同名のメタデータの値をデフォルト値とする（あるメタデータの値を持つデータを検索したい場合、そのメタデータが付与されていないデータについては、そのデータが入っているディレクトリのメタデータの値をそのデータのメタデータの値とみなして検索する）ように実装できる。その結果、例えば同じ日付のデータを複数入れたディレクトリの場合、日付のメタデータはディレクトリにだけ付いており、個々のデータには付かないので、記憶容量を節約することが出来る。またこの方式では、ディレクトリ内の大半のデータがデフォルトの日付をメタデータとする時に、そのディレクトリに例外的な日付をメタデータの値として持つデータを入れることも妨げないように出来る。（検索の際には、メタデータとして日付を持たないデータはデフォルトの日付として照合し、メタデータとして日付を持つデータはそのデータに付与されている日付で照合する）。

【0085】また、本発明の実施形態としては、メタデータ登録装置全体をWEBサーバやファイルシステムなどのデータサーバ装置の一部として組み込む方式の他に、メタデータ登録装置の一部の機能をデータサーバの外部に独立させる実施方式がある。

【0086】図13は、メタデータ登録装置の機能の中で、メタデータ生成に関する機能を全て外部に独立させた実施例である。この構成では、登録するデータの型を検出し（201）、その型に対応するメタデータ生成手続きを選択し（202）、選択した手続きを使ってメタデータを生成し（203）、その結果を例えば図9に示すような形式でデータサーバ装置400に返す機能を、データサーバ装置400とは独立したメタデータ抽出部402として実施している。データサーバ装置400内のメタデータ生成呼出部401が、メタデータ抽出部4

02にデータを渡し、その結果として生成されたメタデータを受け取り、データサーバ装置400内のメタデータ登録部403が、受け取ったメタデータをデータ管理部102経由で記憶装置103に登録する。データサーバ装置400とメタデータ抽出部402の間は、プロセス間通信やネットワークを介した通信によって実現する。

【0087】図14は、メタデータ登録装置の機能の中で、特定の型のデータに対するメタデータを生成する機能を外部に独立させた実施例である。この構成では、データサーバ装置500から与えられた特定の型のデータに対するメタデータを生成し（503）、その結果を例えば図9に示すような形式でデータサーバ装置に返す機能を、データサーバ装置500とは独立したメタデータ抽出部502として実施している。データサーバ装置500内のメタデータ生成呼出モジュール501が、メタデータ生成手続き選択モジュール202により選択した生成手続きがデータサーバ装置500の外部で実行すべきものである場合には、メタデータ抽出部502にデータを渡し、その結果として生成されたメタデータを受け取る。メタデータ抽出部502は、データの型に応じて複数存在して良い。メタデータ生成呼出モジュール501は、複数のメタデータ抽出部502のうち、メタデータ生成手続き選択モジュール202から指示された手続きを持つメタデータ抽出部に、データを渡すことになる。

【0088】図14のような方式を実施する場合には、メタデータ生成手続き選択モジュール202の参照するメタデータ生成手続き表の該当する型のメタデータ生成手続き項目に、そのデータの型のメタデータを抽出する外部の機能を呼び出すために必要な情報を記録しておく。例えばUNIXのプロセス間通信機能であるSOCKETを使ってデータサーバ装置500とメタデータ抽出部502が通信する場合には、SOCKETのアドレスを記録しておく。図14に示した実施例と、図6に示した実施例とは、併用して実施することも可能である。

【0089】以上詳述した実施形態はあくまで一例であり、本発明はその趣旨を逸脱しない範囲で種々の変形実施が可能である。

【0090】

【発明の効果】以上説明したように、本発明によれば、登録するデータに自動的にメタデータを付与することができるので、個々の登録データ毎にユーザがメタデータを付けるような煩雑な作業を軽減することができ、必要なメタデータを確実に付与することができる。

【0091】また、同じ型のデータにはいつも同じ名前でメタデータを付けることが出来、さらに、違う型のデータでも、メタデータ生成手続きのレベルで同じ意味を持つメタデータには同じ名前をつけるように合わせておくことができ、幅広いデータに対して同じメタデータ名

を指定した検索を可能にする効果がある。

【図面の簡単な説明】

【図1】 WEBサーバのデータ管理法を示す図。

【図2】 UNIXのファイルシステムのデータ管理法を示す図。

【図3】 WebDAVのデータ管理法を示す図。

【図4】 データサーバ装置の構成例を示す図。

【図5】 本実施形態に係るメタデータ登録装置の構成例を示す図。

【図6】 本実施形態に係るメタデータ登録装置を備えたデータサーバ装置の構成例を示す図。

【図7】 メタデータ生成手続き表の例を示す図。

【図8】 メタデータ登録処理の流れを示すフローチャート。

【図9】 生成したメタデータの例を示す図。

【図10】 XML文書の例を示す図。

【図11】 XMLメタデータ対応表の例を示す図。

【図12】 XMLデータから生成したメタデータの例を示す図。

【図13】 本実施形態に係るデータサーバ装置からメタデータ生成機能を独立させた場合の構成例を示す図。

【図14】 本実施形態に係るデータサーバ装置から特定のデータ型に対応するメタデータ生成機能を独立させた場合の構成例を示す図。

【符号の説明】

201…データ型検出モジュール

202…メタデータ生成手続き選択モジュール

203…メタデータ生成モジュール

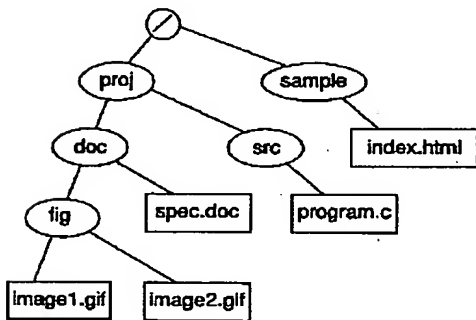
204…メタデータ登録モジュール

301…アクセス要求処理部

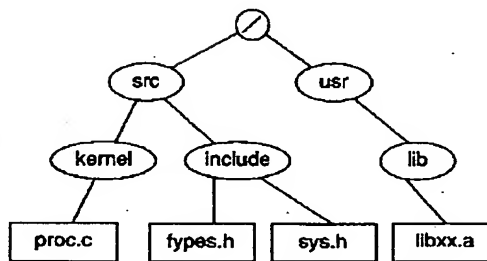
302…データ管理部

303…記憶装置

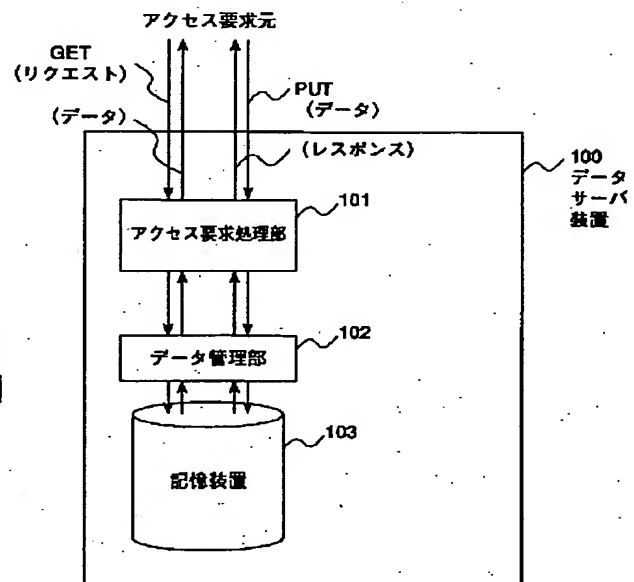
【図1】



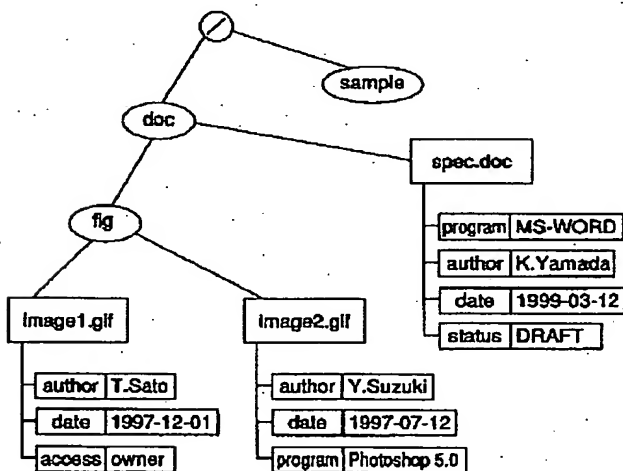
【図2】



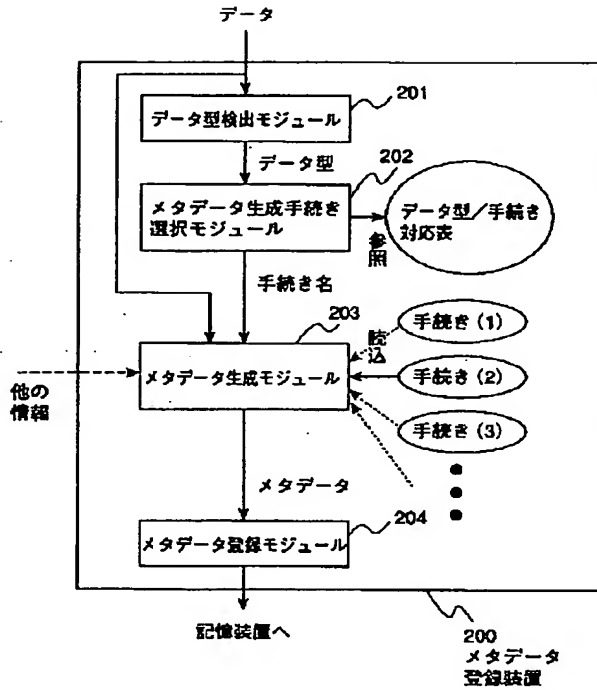
【図4】



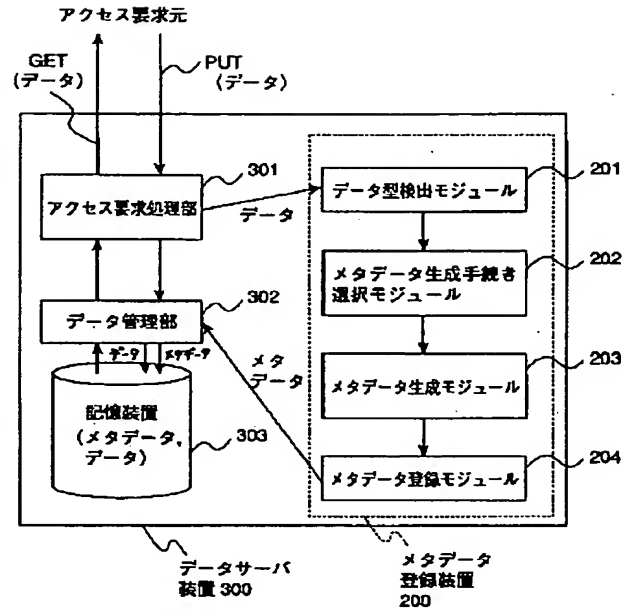
【図3】



【図5】



【図6】



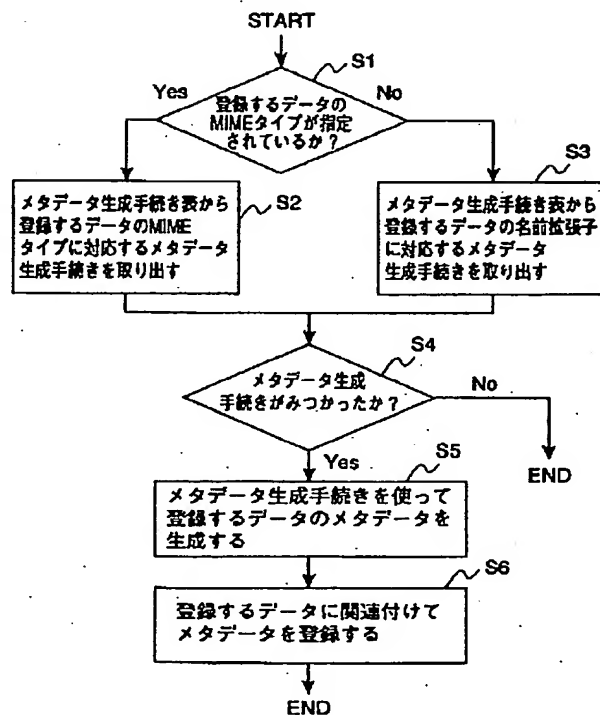
【図7】

MIMEタイプ	拡張子	メタデータ生成手続き
application/msword	doc	metadoc
application/pdf	pdf	metapdf
image/jpeg	jpeg,jpg	metajpeg
image/tiff	tiff,tif	metatiff
video/x-msvideo	avi	metaavi
video/quicktime	qt.mov	metaqt
text/xml	xml	metaxml
...

【図9】

date	1998-07-12
author	Y.Suzuki
program	photoshop 4.0

【図8】



【図10】

```

<?xml version="1.0">
<!DOCTYPE book SYSTEM "book.dtd">
<book>
  <title>Practical File System Design with the Be File System</title>
  <author>Domanic Giampaolo</author>
  <publisher>Morgan Kaufmann Publishers, Inc.</publisher>
  <isbn>1558604979</isbn>
  <date>November 1998</date>
  <price unit="USD">34.95</price>
  <page>237</page>
  <dimension unit="inch">0.62 x 8.97 x 7.04</dimension>
  <contents>
    <chapter num="1">Introduction to the BeOS and BFS</chapter>
    <chapter num="2">What Is a File System</chapter>
    <chapter num="3">Other File Systems</chapter>
    <chapter num="4">The Data Structures of BFS</chapter>
    <chapter num="5">Attributes, Indexing, and Queries</chapter>
    <chapter num="6">Allocation Policies</chapter>
    <chapter num="7">Journaling</chapter>
    <chapter num="8">The Disk Block Cache</chapter>
    <chapter num="9">File System Performance</chapter>
    <chapter num="10">The Vnode Layer</chapter>
    <chapter num="11">User-Level API</chapter>
    <chapter num="12">Testing</chapter>
    <chapter num="Appendix">A File System Construction Kit</chapter>
    <chapter>Bibliography</chapter>
    <chapter>Index</chapter>
  </contents>
  <keywords>BeOS, File System, Journaling, Index</keywords>
</book>

```

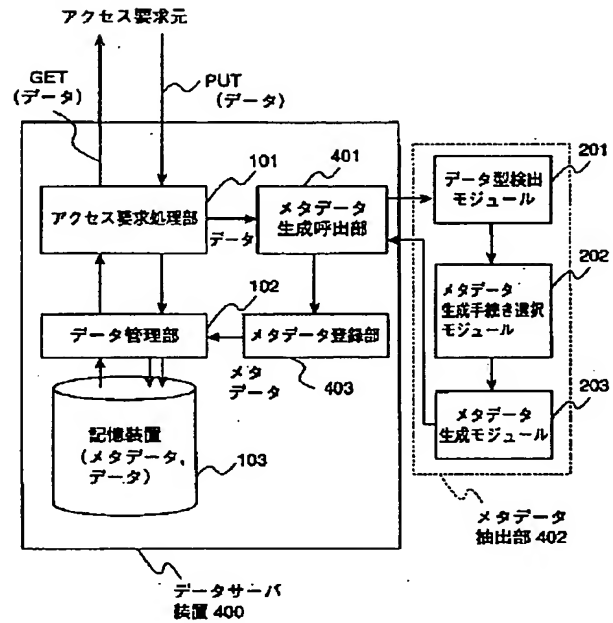
【図12】

title	Practical File System Design with the Be File System
author	Domanic Giampaolo
keyword	BeOS, File System, Journaling, Index

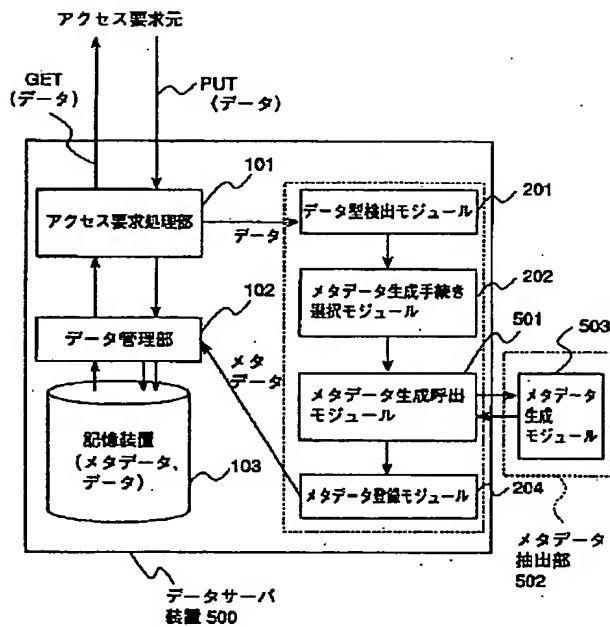
【図11】

DTD名	エレメント名	メタデータ名
book.dtd	title	title
book.dtd	author	author
book.dtd	keywords	keyword
photo.dtd	title	title
photo.dtd	GPS info	location
photo.dtd	date	date
photo.dtd	creator	author
...

【図13】



【図14】



フロントページの続き

(72) 発明者 前田 誠司
神奈川県川崎市幸区小向東芝町 1 番地 株
式会社東芝研究開発センター内
(72) 発明者 横川 健
神奈川県川崎市幸区小向東芝町 1 番地 株
式会社東芝研究開発センター内
(72) 発明者 矢尾 浩
神奈川県川崎市幸区小向東芝町 1 番地 株
式会社東芝研究開発センター内

(72) 発明者 鳥井 修
神奈川県川崎市幸区小向東芝町 1 番地 株
式会社東芝研究開発センター内
(72) 発明者 矢野 浩邦
神奈川県川崎市幸区小向東芝町 1 番地 株
式会社東芝研究開発センター内
(72) 発明者 田中 久子
神奈川県川崎市幸区小向東芝町 1 番地 株
式会社東芝研究開発センター内
Fターム(参考) 5B075 NK02 NK13 NK31 NR05
5B082 EA01 EA07